OPEN DATA CENTER ALLIANCE℠:
IMPLEMENTING THE OPEN DATA CENTER ALLIANCE VIRTUAL MACHINE INTEROPERABILITY USAGE MODEL

TABLE OF CONTENTS

OPEN DATA CENTER ALLIANCE^{SM}: IMPLEMENTING THE OPEN DATA CENTER ALLIANCE VIRTUAL MACHINE INTEROPERABILITY USAGE MODEL

EXECUTIVE SUMMARY

Enterprise IT operators derive three main benefits from the adoption of cloud technology:

1. Resource "arbitrage"
2. Relatively small units for resource allocation
3. Metered resources

In the financial world arbitrage involves the simultaneous negotiation of the same instrument in different markets with the goal of profiting from price differences. In a cloud environment an IT operator has a choice of running workloads in different geographic locations or from services delivered by a variety of providers.

In organizations using physical allocation, a new server represents a three-year commitment or longer, and therefore budgeting and procurement are commensurately heavy processes. Securing these services through an internal or external cloud provider is potentially orders of magnitude faster.

Cloud resources are delivered through a service model where the service's performance and cost are specified contractually through a service level agreement. This environment encourages a data driven approach to resource management, allowing for much agile IT processes than those seen under more traditional budgeting models.

Most cloud based applications today run in virtualized environments. A necessary condition for realizing the benefits is a capability for moving workloads across machines. These machines can be part of a private cloud or in a public cloud, and with relocating workloads there is a need to move virtual machines within and across different clouds. This capability is described in an Open Data Center Alliance (ODCA) usage model document[1]. A team led by T-Systems Telekom Innovation Laboratories, the FZI research team from the University of Karlsruhe and supported by Intel Corporation carried out a proof of concept (PoC) project to implement the usages described in the document, described in this report.

[1] *Open Data Center Alliance Usage: Virtual Machine (VM) Interoperability in a Hybrid Cloud Environment*

In context of the ODCA VM Interoperability Usage Model mentioned, an initial selection was made of commonly found products in the corporate cloud environment, and a set of appropriate experiments defined to test them against the Usage Model. The results shown in this document demonstrate how closely these potential systems aligned to the Usage Model at the time of the PoC tests towards demonstrating Cloud technology maturity. Later iterations of testing may expand the scope of products as far as practical for a PoC test, and as software versions evolve. Therefore the results shown represent a minimum suggested baseline for cloud expectations and current cloud technology state-of-the-art, which should significantly improve for every subsequent test which the reader may wish to initiate.

THE VM INTEROPERABILITY PROJECT

The goal for this project was to devise and implement testing methods and conduct a basic set of functional tests relevant to the ODCA Usage for Virtual Machine (VM) Interoperability in a hybrid cloud environment using both private and public clouds.

Ideally, cloud subscribers would like to be able to select any cloud provider based on basis of service cost, performance and capabilities. They also want to link, per need, private clouds made up of dedicated services with public clouds that consist of shared services. In order to make this feasible for the cloud consumer, the various hypervisor platforms and virtual machines (VMs) involved will need to be interoperable and enable portability, leveraging defined industry standards.

As cloud service scenarios evolve in the delivery of IT, (including business, public and private applications), those organizations recognize the execution of IT needs to evolve to realize the benefits of cloud technology. Some of those changes may include the need to actually move systems between cloud locations and in order to achieve this, the ability must exist for these cloud based services to be quantified, packaged, and transferred to their new destination with some degree of reliability.

Reliability and reproducibility of a change such as a VM migration are based on pre-defined standards, specifications, frameworks, scenarios, and processes. The participants and members in the ODCA, with practitioners representing most of the mentioned scenarios above, concur that this need exists in their organizations too for reasons such as being able to demonstrate the ability to move between internal private clouds, being able to move between cloud providers if necessary, and if for no other reason, to demonstrate that the service is not locked in to that environment with no relocation options once it has been established there (i.e. avoid creating “future legacy environments”)

PoC participants agreed to focus on the base parameters and characteristics necessary to enable VM Interoperability according to the ODCA VM Interoperability Usage Model[2].

The project includes the following components:

1) Assembly and deployment of a basic virtualized infrastructure, the project test bed
2) Selection of a number of hypervisor environments and virtualized operating system images
3) Creation of virtual machines and moving the virtual machines across physical hosts. The most commonly found combinations of hypervisor hosts and OS images were exercised and documented.
4) The experiments were conducted to illustrate the three goals described in the ODCA VM interoperability usage document, namely
    - Check interoperability
    - Move or copy between two hypervisors and cloud providers
    - Leverage common operation and interoperability
5) Running the actual tests. The tests were scripted. Even though running each case required a fair amount of time, the exercises were repeated as necessary to insure consistent and valid results.

PoC activities continue at the time of the first release of the findings, now focused on dealing with some of the more complex challenges that were identified, including demonstrating interoperability across public cloud service providers. These challenges will require more time and focus to measure and quantify.

Overall though, it was demonstrated that with the right preparation, VM Interoperability is indeed possible as per the framework provided by the ODCA VM Interoperability Usage Model, and that it forms a valid reference base for further evolution of Cloud Interoperability capability.

VM INTEROPERABILITY OPERATIONAL MODEL AND THE OVF

Deployment and distribution of applications in organizations is undergoing rapid transformation, largely due to the maturing virtualization technology and the increasing role of cloud computing infrastructure. Virtual machine mobility has started to play central role in scaling capacity, disaster avoidance and reducing downtime. Most importantly, it provides end users with the option to choose from a number of cloud service providers where they can deploy their application.

The Distributed Management Task Force (DMTF) recognized challenges associated with the creation of portable virtual machines early on and has been working on an industry standard format for portable virtual machines. With virtual machines packaged in this format, the VMs can be installed on any virtualization platform that supports the

[2] *Open Data Center Alliance Usage: Virtual Machine (VM) Interoperability in a Hybrid Cloud Environment*

standard, simplifying interoperability, security and virtual machine lifecycle management for virtual infrastructures.

The ODCA and DMTF share these common concerns and have announced plans to collaborate to try to resolve interoperability challenges hindering cloud adoption through synthesis of enterprise customer requirements, usage models, recommendations and standardization of application deployment practices. This proof of concept represents one of the first activities fulfilling the joint DMTF and ODCA goals.

The target machines used in the PoC tests are mainstream systems. Although not a primary goal for this work, when the tools allow/support it, the test cases follow the OVF specification in the context of the ODCA VM Interoperability Usage Model.

Conversely, the test cases in this project were conducted under the portability framework defined by the DMTF: For the purposes of this report we define three types for VM migration operations, namely types 1, 2 and 3. These correspond roughly to the three levels of portability defined by the DMTF.[3]

To summarize the DMTF definitions:

Level 1: Workloads under Compatibility Level 1 only run on a particular virtualization product and/or CPU architecture and/or virtual hardware selection. Level 1 portability is logically equivalent to a suspend operation in the source environment and a resume in the target environment.

Level 2: Workloads under Compatibility Level 2 are designed to run on a specific family of virtual hardware. Migration under Level 2 is equivalent to a shutdown in the source environment followed by a reboot in the target environment.

Level 3: Workloads supporting Level 3 compatibility are able to run on multiple families of virtual hardware.

For the work described in this document, Type 1 is also known as *live migration*; Type 2 is *cold migration*, and we name Type 3 *loosely coupled migration* for lack of an established term.

Live migration, when feasible, preserves the most operating state of the three schemes, including IP addresses, and open file descriptors and even transactions and streaming data in midstream. Live migration may be required by some legacy applications that break after some of the state transitions mentioned above. Requirements for live migration are strict: the target host usually needs to be part of a preconfigured cluster; the hosts need to be in the same subnet, and even if physically remote hosts are connected through a VPN tunnel, latency due to the trombone effect may induce failures. Live migration is not possible across heterogeneous hypervisor environments.

[3] DMTF, *Open Virtualization Format White Paper*, OVF version 1.0.0e, paper DSP2017, 2/6/2009, Section 5

Cold migration allows additional operating environment changes compared to live migration. A logical shutdown and restart means that VMs in the target environment may end up running with a different set IP addresses. Open file descriptors may be different, even though they may be reaching the same files, the descriptors may point to a remote file that was previously local. Transactions interrupted during the migration may have to be rolled back and retried. Also, depending on the target environment, the new hard drive may have been converted from the virtual format in the source environment. The work of this PoC focuses on cold migration. For applications that can tolerate these environment changes, cold migration offers the broadest potential for across a number of varied hypervisor environments.

Loosely coupled migration provides the least support of state preservation. At the same time it provides the most options in running an application across computer architectures or service providers. It involves rebuilding the operating environment in the target environment, and may require as little as being able to reconnect to the original storage. This case applies to advanced methodologies, such as integrated development and operations (DevOps)[4]. On the other hand this case also covers applications run in public clouds that can't be moved because of customization in the provider's environment make it difficult to move a virtual machine out of that environment. Although loosely coupled migration offers the broadest choices for operating environments, running not only on a variety of hypervisor environment, but across platform architectures, the number of applications capable of continuing to run under these changes is much smaller than those that can run under the cold migration model.

TEST BED SET UP OVERVIEW

The test bed was hosted in the Intel End-user Integration Center Test Lab Cloud in an isolated network partition consisting of four server nodes and a console computer. T-Systems technical staff accessed this environment through the standard Web portal and associated access applets. The setup provided exclusive, out of band access to the hardware including low level operations down to setting up BIOS sheets and cycling the power in the machines. Figure 2 illustrates the test bed deployment architecture and its components.

The outcome of a test case evaluation gives an overall score describing the success of an interoperability test between two hypervisors. A score is expressed on a three-level nominal scale consisting of the values SUCCESS, WARNING and FAILURE. The score is derived from the whole set of scores assigned to the single tests, determined by predefined evaluation rules. The rules define the score, also the three levels, is assigned to each of the single tests. If we look at the values above as an ordered set, with FAILURE the lowest value, the overall score of the test series is the lowest of any single test. Therefore if any single test fails, the overall test is scored as FAILURE. Likewise if any single test has a WARNING outcome, with the other tests being SUCCESSful, the overall test is scored as WARNING. This mode of scoring is called the aggregation rule.

[4] http://www.readwriteweb.com/enterprise/2011/08/devops-what-it-is-why-it-exist.php, http://www.cio.com/article/print/705919

The bar for the conversion outcomes was set high, and hence any outcome labeled as WARNING could likely result in a runnable machine. For instance, the machine in the new environment may have started with a re-qualified set of IP addresses or a new processor model or failures in firewall and routing rules, which may be a function of the environment defined by the target hypervisor more than a specific issue associated with the conversion. Likewise if a test is re-run and the machines restart with new IP addresses each time, this triggers an aggregation rule WARNING. In spite of these changes, for practical purposes, the machine may end up running just fine in the target environment. Also in some cases the converted virtual hard drive had to be attached to a manually created virtual machine in the target environment. Even if this action yields a runnable machine, it runs counter to the notion of automated migration.

Furthermore, the test sets comprise not only hypervisor level tests but operating system level tests as well. Additional tests allow assessing the success of a VM migration in more detail and increasing the strictness for gaining a positive score. At the operating system level, tests demand an (almost) identical configuration in terms of CPU speed, RAM size, IP address, routing table or DNS settings, to accomplish an evaluation outcome of SUCCESS or WARNING.

TEST BED SOLUTION STACK COMPONENTS

This section describes the technology components used in the project. A snapshot was taken at the beginning of the experiment series. Some of the components have since evolved. However the original mix was preserved during the series of runs for consistency.

No single tool currently exists to perform the VM image conversions across the four hypervisor environments comprehended in the PoC. The technical team literally used a quiltwork of tools to cover the conversion matrix. Some conversions were done directly from the source to target hypervisor environments; some use two tools, with an intermediate conversion to OVF. Some conversions use a standalone tool, and some use conversion tools available from the hypervisor vendor's console. Where more than one method was available to perform a conversion, a reasonable effort was made to try all of them and the best result was recorded.

- Hypervisors
    - VMware ESXi 5.0.0, Citrix Xen Server 6.0.2, KVM (Cent OS6.3), Microsoft Hyper-V (Windows Server 2008 R2)
- Virtual Machine Monitor consoles
    - VMware vSphere Client v5.0.0, Citrix Xen Center 6.0.2, Virtual Machine Manager 0.9.4 for KVM , Microsoft Hyper-V Manager 6.1
- VM Converters & tools:
    - Citrix XenConvert 2.3.1, VMWare vCenter Converter Standalone 5.0.0, virt-convert, virt-v2v, quemu-img
- Guest OS:
    - CentOS6.3 64 bit, Ubuntu12.04 64 bit, Microsoft Windows Server 2008 R2 64 bit

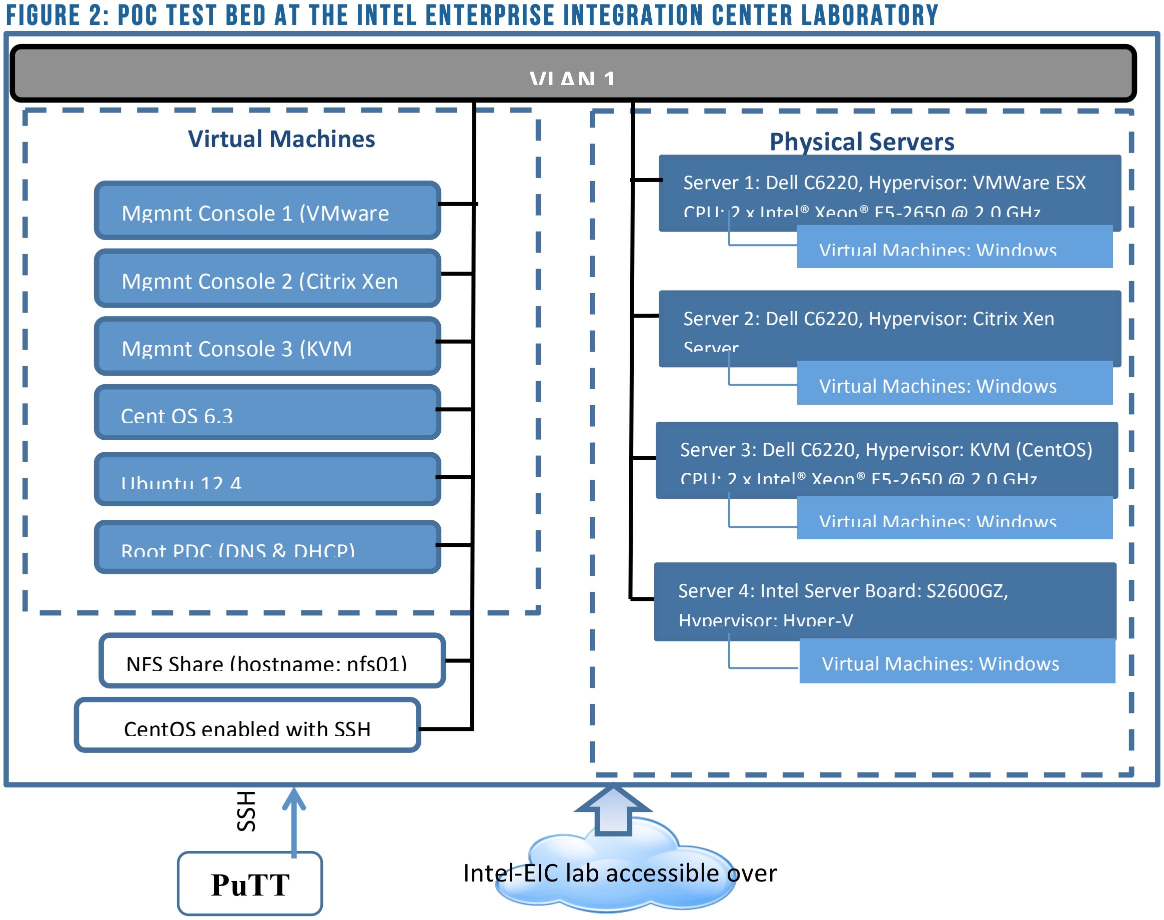
HARDWARE COMPONENTS

The following hardware was deployed in the lab:

- Three Dell C6220 Servers each with dual Intel® E5-2650 processors running at 2.00 GHz and provisioned with 64GB RAM and a 280 GB hard disk.
- One Intel® white box server with Intel® Server Board S2600GZ with dual Intel® E5-2850 processor running at 2.7 GHz and provisioned with 64GB RAM and a 1 TB hard disk.
- Network switches and router as needed
- Virtual Machines to function as management consoles
- Two NFS shares with 800 GB and 400GB storage

The microprocessors used are Intel® Xeon® E5 series on the servers hosting the hypervisors. While the CPUs are identical for VMware, Xen and KVM, it is different for Hyper-V. The detailed PoC test methodology process has scripts running inside the VMs that reports CPU information on the source and target hypervisors along with other metrics. In the test results with Hyper-V the change in CPU were correctly reported after VMs were migrated.

FIGURE 2: POC TEST BED AT THE INTEL ENTERPRISE INTEGRATION CENTER LABORATORY



EXPERIMENTAL RESULTS

This section summarizes the results for the interoperability tests. Each test was scripted to execute the following operations:

1) A sanity check at the source hypervisor environment to ensure readiness for migration
2) Shutting down the VM in the source hypervisor environment
3) Exporting the VM to OVF or to destination format as applicable
4) A resource check in the target hypervisor environment
5) A consistency check in the target hypervisor environment to determine if the VM is runnable
6) Importing the VM into the target hypervisor environment
7) Restarting the VM in the target hypervisor following the shutdown in step 2

The matrixes below summarize the overall results. Under the aggregation rule a conversion is given a fail if any of the seven steps above fails or cannot be completed. The tools used to carry out the VM conversions are not disclosed in this document. The goal for this research was to provide a broad indication of the state of the art for VM interoperability and not to carry an exercise in ratings tools. Although reasonable care was taken to give each tool a fair shot at performing VM conversions, the team could not rule extraneous factors, using the tool in a manner the vendor did not intend, and the fact that taking the initial snapshot would mean that some of the tools would be a revision behind by the time this research was published. In any case there was an effort to communicate the results privately with each vendor.

It is an interesting fact to note that the success rate varies depending on the OS image run on the VM, even though the same tool was used to perform any given conversion. Windows Server 2008 was the most forgiving; CentOS the least with Ubuntu somewhere in between. The results reflect the methodology used in the experiments, and should not construed to indicate the level of support for interoperability of any operating system.

WARNING outcomes were mostly the VM reporting changes in IP addresses or changes in memory configuration and CPU speed. In some cases a VM had to be created in the target environment to run the virtual hard disk drive in the target environment.

Most failures were due to the VM inability to acquire IP addresses in the new environment. This could have been caused by the migrated machine's inability to connect to the DHCP service in the new environment (possibly due to a minor problem in the network portion of the migrations). However this kind of individual investigation was beyond the scope work more intent in demonstrating processes for interoperability.

In context of the ODCA VM Interoperability Usage Model, the research team selected a set of publicly available conversion tools. The tables following represent a set of results demonstrating how closely these potential systems aligned to the Usage Model PoC experiments. Later iterations of testing may expand the scope of products as far as practical for a PoC test, for instance by including interoperability across public clouds, and as software versions evolve, significant improvement in interoperability outcomes are expected. These tables therefore represent a baseline against the ODCA usage models.

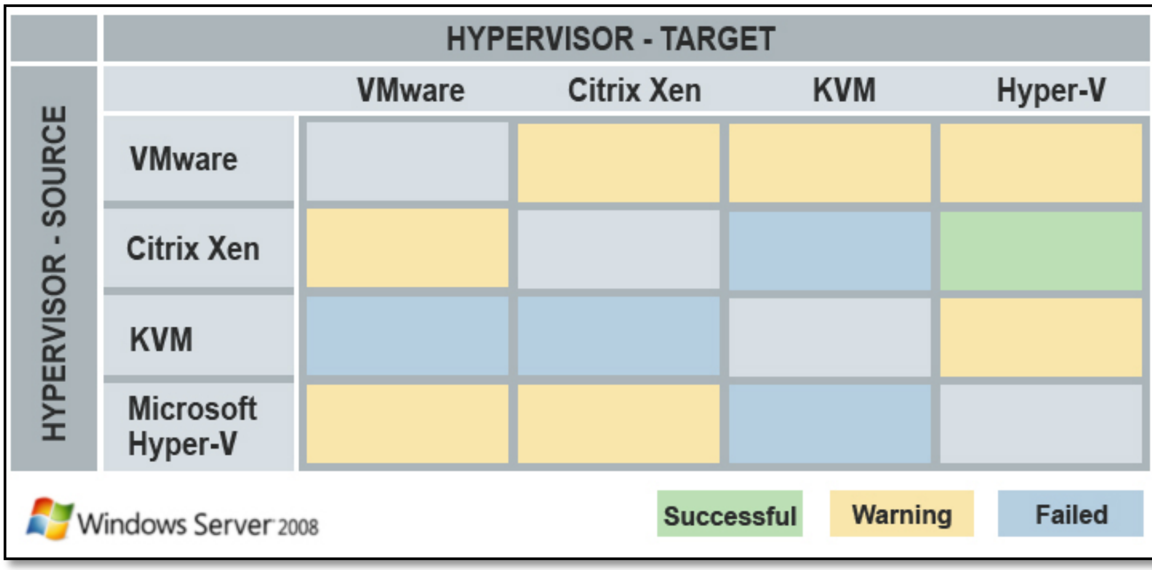


FIGURE 3: SUMMARY RESULTS FOR A WINDOWS SERVER 2008 IMAGE.

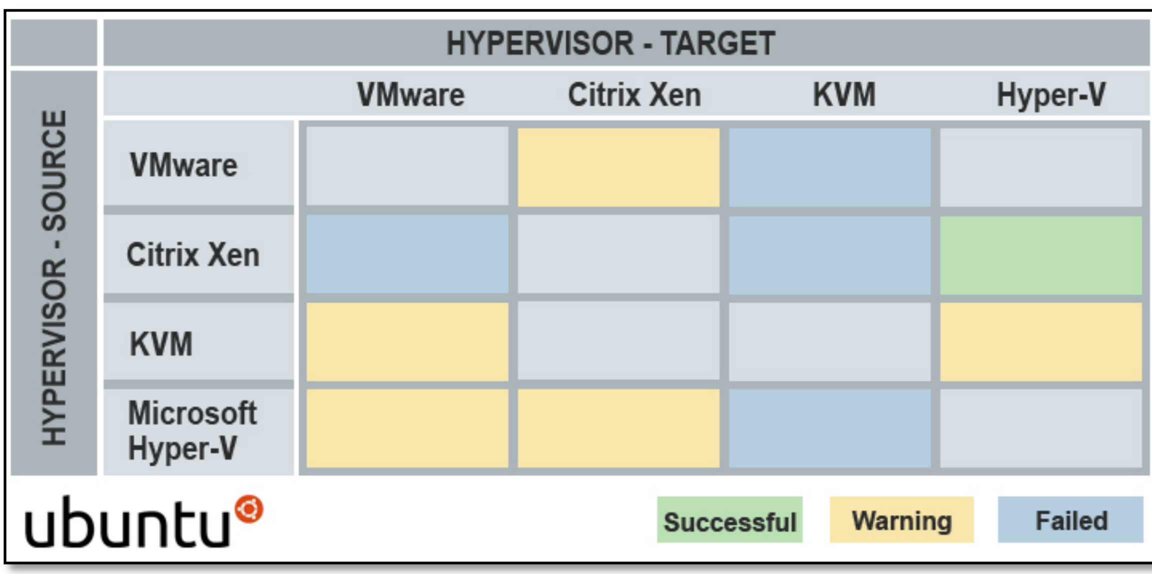


FIGURE 4. SUMMARY RESULTS FOR AN UBUNTU 12.04 IMAGE

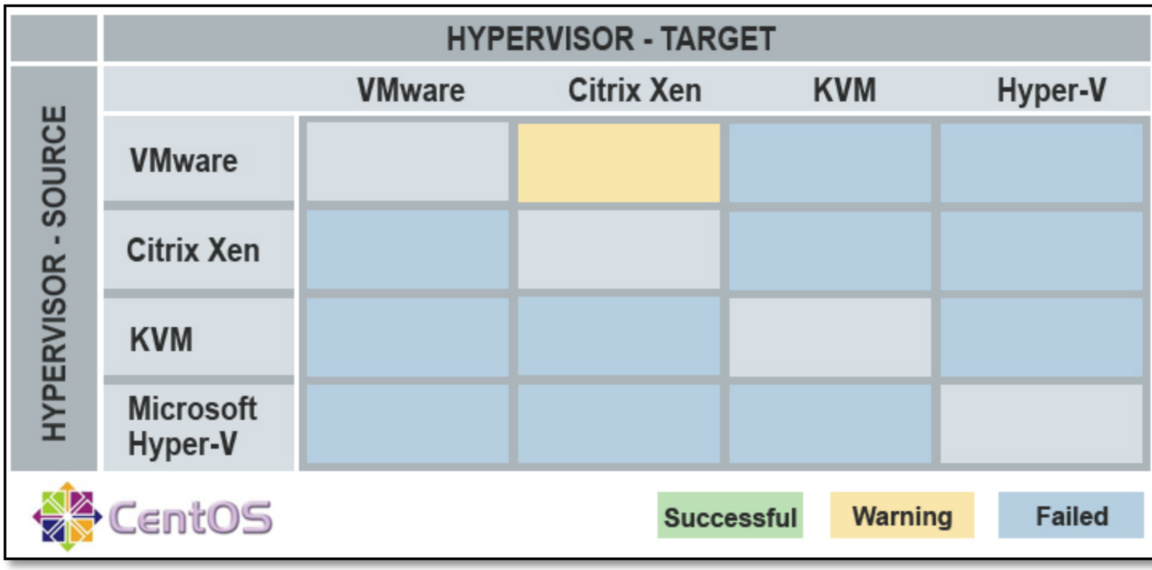


FIGURE 5. SUMMARY RESULTS FOR A CENTOS 6.3 IMAGE

CONCLUSIONS

A capability for VM interoperability is an important precondition to truly realize the oft expressed benefits of virtualized clouds, such as the ability to balance resources through fungible pools of resources, business continuity and load balancing by leveraging distributed publicly available resources, as well as demonstrable avoidance of lock in to a single Cloud Provider, platform or technology.

To the knowledge of the participants of this project, this experiment is the first of its kind published to a broad audience, leveraging a broad spectrum of hypervisor environments, guest operating systems and publicly available conversion tools (typically those bundled with the hypervisor or generally available to the provider who uses the hypervisor and selected test operating systems).

For conversions reported as WARNING the team chose to err on the side of rigor, and for instance changes in resource memory size or IP Address changes or firewall and routing rules or inability pause and un-pause the migrated VMs were flagged as warnings. Most applications will continue running under these circumstances, and therefore it might be possible to relax some of these requirements. For instance, if an application is known to be impervious to these changes, the operator may select to ignore these changes. However, the team recommends that these conditions should be relaxed only after a thorough discussion in the industry and consensus has been built about what the actual practice should be.

The PoC results indicate that real interoperability based on the OVF specification is potentially at an early stage between vendors, that is, not yet very mature in the actual deployments and integration in the hypervisor layer, and in the migration tool layer.

Posted results notwithstanding, given certain scenarios and methodologies, VM Interoperability is indeed possible already. However, with the current state of the art companies need to be prepared to accept certain limited manual interactions now to improve interoperability outcomes as an interim. It must be noted that for large scale migrations, this may be risky, and potentially less acceptable in the context of contractually orientated migrations.

Particular care was given to document interoperability across heterogeneous hypervisor environments, reflecting the ODCA goal of maximizing the service options for its members.

An open question to the industry is considering using an alternate approach to packaging machines for interoperability rather than OVF such as CIM CIMI using catalogs and brokering between catalogs.

Interoperability is an absolutely essential dimension for cloud services to help to avoid corporations being locked in, and that solution alternatives are always available, for any reason, be it for service scalability, security, reliability or any other business reason. Interoperability also facilitates trialing of solution alternatives including in house options with a variety of service providers objectively and with a minimum of inconvenience.

The results of the PoC DO NOT indicate “success” or “failure” of cloud interoperability. In the context of this PoC and the associated Usage Model (VM Interop), they should be considered as an indicator of the state of technology in context of VM Interoperability at the time of testing. Every new software version and update to the industry standards and specifications (such as OVF) is rapidly moving the potential results in the direction of improved service interoperability outcomes as specified by the service contract or SLA. This means that the tests described in this document are valid at the time of testing, and establish a good baseline, but any party wishing to conduct an interoperability activity for any business reason should repeat the tests with the current technology and selected relevant components therefore advancing the snapshot to the updated components. The results will constantly improve from this baseline now established, in context of the VM Interoperability Usage Model, and they should only expect constantly improving results from this point forwards.

To the knowledge of the participants of this project, this experiment is the first of its kind in the public domain, leveraging a broad spectrum of hypervisor environments, guest operating systems and publicly available conversion tools.

The project participants would like to encourage a healthy dialog in the industry to advance the state of the art to the point that interoperability becomes a second order consideration, allowing users to focus on the business problems at hand instead.

